УДК 519.8

И.В. Козин
Запорожский национальный университет МОН Украины, Украина
Украина, 69600, г. Запорожье, ул. Жуковского, 66

Эволюционная модель задачи булева программирования

I.V. Kozin
Zaporizhzhya National University MES of Ukraine, Ukraine
Ukraine, 69600, c. Zaporizhzhy , Zhukovsky str., 66

Evolutionary Models Boolean Programming Problem

І.В. Козін
Запорізький національний університет МОН України, Україна
Україна, 69600, м. Запоріжжя, вул. Жуковського, 66

Еволюційна модель задачі булева програмування

В работе представлены результаты исследования фрагментарной и эволюционной модели задачи булева программирования. Показано, что при определенных условиях задача булева программирования может рассматриваться как задача на фрагментарной структуре. Предложена эволюционно-фрагментарная модель задачи на множестве перестановок с геометрическим оператором кроссовера. Метод протестирован на наборе индивидуальных задач различных размерностей.
Ключевые слова: задача о рюкзаке, фрагментарная структура, эволюционная модель.

The results of the study fragmented and evolutionary model Boolean programming problem. It is shown that under certain conditions, the problem of Boolean programming can be seen as a problem in the fragmented structure. Proposed evolutionary model of the fragmented on the set of permutations with geometric crossover operator. Method is tested on a set of individual tasks of various dimensions.
Key words: knapsack problem, fragmented structure, evolutionary model.

У роботі представлені результати дослідження фрагментарної і еволюційної моделі задачі булева програмування. Показано, що при певних умовах задача булева програмування може розглядатися як задача на фрагментарній структурі. Запропоновано еволюційно-фрагментарна модель задачі на множині перестановок з геометричним оператором кросоверу. Метод протестований на наборі індивідуальних задач різних розмірностей.
Ключові слова: задача про рюкзак, фрагментарна структура, еволюційна модель.

Введение

Во многих прикладных оптимизационных задачах отыскание точного решения является достаточно трудной проблемой. Иногда единственным методом, который гарантирует желаемый результат, является лишь полный перебор всех допустимых решений. Аналогичная проблема возникает при отыскании точного решения многих задач распознавания, в частности при отыскании решения задач, для которых доказано свойство *NP*-полноты.

Для поиска решения подобных задач, как правило, используются приближенные и эвристические алгоритмы. Одним из эвристических подходов является подход, основанный на применении фрагментарных алгоритмов [1], [2]. Однако использовать

фрагментарный алгоритм для поиска оптимального решения задач с четко определенным формальным критерием вряд ли разумно. Фрагментарный алгоритм останавливается при отыскании первого максимального по включению фрагмента и, следовательно, не гарантирует качества решения с точки зрения оптимума критерия. Для решения дискретных оптимизационных задач оправданным является метод, основанный на комбинации эволюционного и фрагментарного алгоритмов. В настоящей работе подобная технология будет применена к известной задаче булева программирования – задаче о рюкзаке [3].

Постановка задачи. Задачу о рюкзаке будем рассматривать в следующей постановке: на множестве n-мерных двоичных векторов

$$Z_2^n = \{x \mid x = (x_1, x_2, ..., x_n),\ \ x_i \in \{0,1\}, i = 1,2,...,n\}$$

найти вектор $x^* = (x_1^*, x_2^*, ..., x_n^*)$, для которого значение линейной целевой функции

$$F(x) = c_1 x_1 + c_2 x_2 + ... + c_n x_n \tag{1}$$

максимально при условии выполнения m линейных неравенств

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \le b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \le b_2 \\ ...... \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \le b_m \end{cases} \tag{2}$$

Коэффициенты c_i, a_{ij}, b_j – неотрицательные вещественные числа, $i = 1,2,...,n,$ $j = 1,2,...,m.$

В [4] доказано, что задача о ранце является NP-полной. Одним из методов отыскания точного оптимального решения этой задачи является метод ветвей и границ, описание которого можно найти во многих источниках [3], [4].

Простой, широко известный жадный алгоритм отыскания приближенного решения задачи о рюкзаке заключается в следующем: для каждой переменной x_i, i=1,2,…,n вычисляем ее удельную ценность

$$y_i = \min_{a_{ji} > 0,\, j=1,2,...m} \{c_i \frac{b_j}{a_{ji}}\}$$

и упорядочиваем переменные x_i в порядке убывания удельной ценности $x_{i_1}, x_{i_2}, ..., x_{i_n}$.

На начальном шаге полагаем $x^0 = (0,0,....,0)$. На каждом последующем шаге определяется значение координаты i_k в решении. Изменим решение x^{k-1}, положив координату x_{i_k} равной 1. Если измененное решение удовлетворяет всем ограничениям (2), то полагаем x^k равным этому измененному решению. В противном случае оставляем $x^k = x^{k-1}$.

Конечно, приведенный алгоритм не гарантирует точности полученного с его помощью решения, а является лишь простой эвристикой, которая позволяет найти некоторое допустимое решение задачи. Попытаемся обобщить эту эвристику с помощью понятия фрагментарной структуры.

Фрагментарная структура

Определение 1. Фрагментарной структурой (X, E) на конечном множестве X называется семейство его подмножеств $E = \{E_1, E_2, ..., E_n\}$, такое, что $\forall E_i \in E, E_i \neq 0 \ \exists e \in E_i, E_i \setminus \{e\} \in E$.

Элементы из множества E будем называть допустимыми фрагментами. Таким образом, для любого допустимого фрагмента E_i существует нумерация его элементов $E_i = \{e_{i1}, e_{i2}, ..., e_{is_i}\}$, такая, что $\forall k = 1, 2, ..., s_i \quad \{e_{i1}, e_{i2}, ..., e_{ik}\} \in E$.

Определение 2. Одноэлементные множества, которые являются допустимыми фрагментами, будем называть элементарными фрагментами.

Определение 3. Фрагмент называется максимальным, если он не является подмножеством никакого другого фрагмента.

Очевидны следующие свойства фрагментов:

Свойство 1. Пустое множество является фрагментом, $\varnothing \in E$.

Свойство 2. Пусть $\max\limits_{i=\overline{1,n}} |E_i| = M$. Тогда в E найдутся элементы, мощности которых будут $M, M-1, M-2, ..., 0$.

Теорема 1. Если (X, E) – фрагментарная структура на множестве X, то для любого непустого множества $A \in E$ существует нумерация его элементов $A = \{x_1, x_2, ..., x_n\}$ такая, что $\forall k, k = \overline{1, n}$ множество $\{x_1, x_2, ..., x_k\} \in E$.

Доказательство. Пусть $A_0 = A$, $A \in E$ и $|A| = n > 0$. Выберем элемент $x_n \in A_0$ таким образом, чтобы $A_1 = A_0 \setminus \{x_n\} \in E$. Повторив эту процедуру $n - 1$ раз по отношению ко множествам $A_1, A_2, ..., A_{n-1}$, получим требуемую нумерацию элементов множества A.

Из теоремы вытекает, что всякий допустимый фрагмент можно построить из пустого множества, последовательно добавляя к нему элементы так, чтобы на каждом шаге такой процедуры полученное подмножество было допустимым фрагментом.

Максимальный фрагмент может быть построен с помощью следующего «жадного» алгоритма:

а) элементы множества X линейно упорядочиваются;

б) на начальном шаге выбирается пустое множество $X_0 = \varnothing$;

в) на шаге с номером k+1 выбирается первый по порядку элемент $x \in X \setminus X_k$, такой, что $X_k \bigcup \{x\} \in E$;

г) алгоритм заканчивает работу, если на очередном шаге не удалось найти элемент $x \in X \setminus X_k$ с требуемым свойством.

Приведенный выше алгоритм построения максимального фрагмента во фрагментарной структуре будем называть фрагментарным алгоритмом. Результат применения фрагментарного алгоритма определяется заданным линейным порядком на множестве X. Таким образом, любой максимальный фрагмент может быть описан некоторой перестановкой элементов множества X. Пусть $A \in E$. Условие для элемента $x \in X$, при котором $A \bigcup \{x\} \in E$, будем называть условием присоединения элемента x.

Теорема 2. Если $A \in E$ и $\forall x \in X$ существует алгоритм полиномиальной трудоемкости по числу элементов множества X для проверки условия присоединения, то задача построения максимального фрагмента является полиномиально разрешимой.

Доказательство. Пусть трудоемкость проверки условия присоединения составляет $O\left(n^k\right)$, где k – натуральное число. Трудоемкость упорядочения элементов может быть оценена величиной $O(n \ln n)$. Тогда для фрагментарного алгоритма построения максимального фрагмента трудоемкость оценивается величиной $O\left(n \ln n + n^{k+1}\right)$, то есть $O\left(n^{k+1}\right)$.

Покажем теперь, что задача булева программирования в постановке (1) – (2) является задачей на фрагментарной структуре. Допустимым фрагментом в этой структуре является такой набор индексов $I \subseteq X = \{1, 2, ..., n\}$, для которого

$$\sum_{i \in I} a_{ji} \le b_j \qquad j = 1, 2, ..., m.$$

Очевидно, что набору индексов I соответствует допустимое решение задачи (1) – (2)

$$x = (x_1, x_2, ..., x_n), \text{ где } x_i = \begin{cases} 1, & i \in I \\ 0, & i \notin I \end{cases} \qquad i = 1, 2, ..., n.$$

Значение целевой функции на этом решении $F(x) = \sum_{i \in I} c_i$. Это же значение будем рассматривать как вес соответствующего набора индексов.

Таким образом, задача булева программирования может быть сформулирована, как задача на фрагментарной структуре $(X, \{I\})$: найти допустимый фрагмент максимального веса. Очевидно, оптимальное решение этой задачи можно искать среди максимальных фрагментов. Если $\forall i = 1, 2, ..., n \;\; c_i > 0$, то любое оптимальное решение задачи может быть получено фрагментарным алгоритмом при соответствующей перестановке элементарных фрагментов. Однако число перестановок фрагментов в модели есть $n!$ и потому задача перебора всех фрагментов является трудной. Для быстрого поиска решений, которые близки к оптимальному, предлагается следующая эволюционная модель на фрагментарной структуре.

Эволюционная модель

Эволюционные (генетические) алгоритмы подробно рассматривались в многочисленных публикациях [5], [6]. Для ряда оптимизационных задач удалось предложить достаточно эффективные процедуры поиска оптимальных решений, основанные на применении эволюционных алгоритмов. Для реализации эволюционного алгоритма необходимо выделить ряд объектов и процедур, совокупность которых будем называть эволюционной моделью [7]. Основные составляющие эволюционной модели следующие:

– базовое множество решений – множество допустимых решений X, на котором ищется оптимальное решение задачи;

– оператор построения начальной популяции: процедура, которая позволяет выделить на множестве всех допустимых решений его подмножество $Y \subseteq X$ для последующей эволюции;

– критерий селекции – алгоритм, который позволяет сравнивать по качеству решения в рамках заданной популяции;

– оператор кроссовера $K : X \times X \to X$, позволяющий по двум допустимым решениям-родителям построить новое решение-потомок из множества допустимых решений;

– оператор мутации $M : X \to X$;

– оператор отбора, который выделяет множество пар в Y для выполнения операции кроссовера;

– оператор эволюции, позволяющий строить новые популяции из множества родителей и потомков;

– правило остановки, которое определяет условие остановки эволюционного алгоритма.

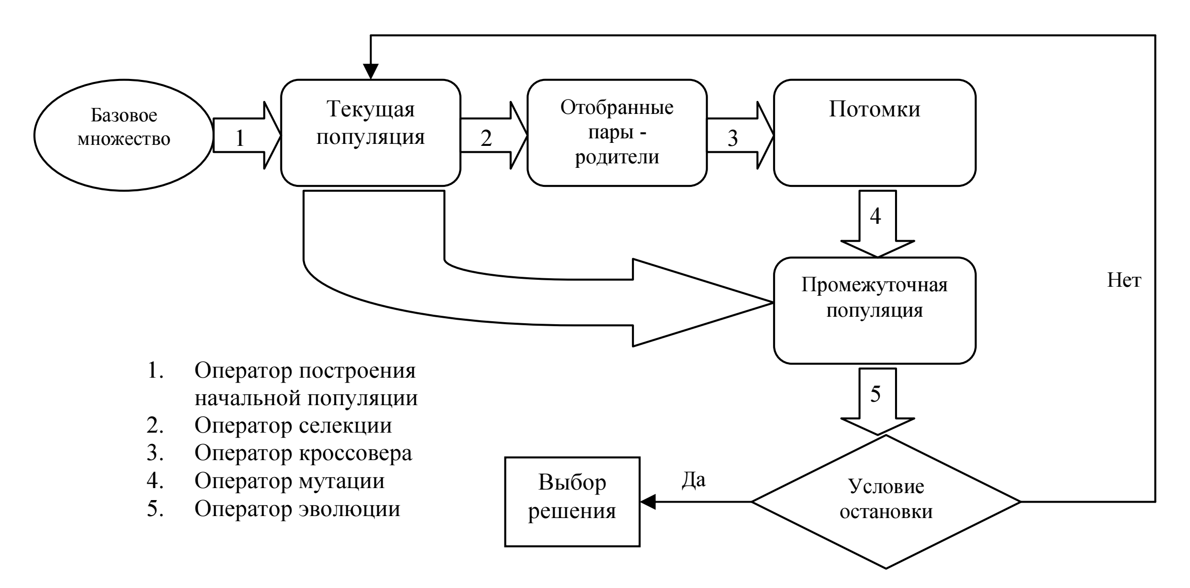


Рисунок 1 – Эволюционная модель

Опишем кратко принцип работы эволюционного алгоритма. На начальном шаге с помощью оператора начальной популяции строится множество решений Y_0. На каждом очередном шаге предполагается заданным некоторое множество допустимых решений – текущая популяция. На первом шаге это множество $Y = Y_0$. Для каждого из элементов множества Y вычисляется значение критерия селекции. Далее с помощью оператора отбора в текущей популяции Y выбирается множество пар для кроссовера. К каждой паре из выбранного множества пар применяется оператор кроссовера, а затем к результату кроссовера применяется оператор мутации. Таким путем находится множество элементов – потомков $\tilde{Y}$.

К промежуточной популяции $Y \cup \tilde{Y}$, которая является объединением текущей популяции и множества потомков, применяется оператор эволюции, который выделяет на этом множестве новую текущую популяцию. Процесс эволюции повторяется до тех пор, пока не будет выполнено условие остановки эволюционного алгоритма. Блок-схема эволюционного алгоритма приведена на рис. 1.

Свойства фрагментарных структур позволяют построить особый класс эволюционных алгоритмов на фрагментарных структурах – ЭВФ-алгоритмы.

ЭВФ-алгоритм является комбинацией эволюционного и фрагментарного алгоритма. Опишем эволюционную модель и принцип действия такого алгоритма.

В качестве множества допустимых решений рассматривается подмножество максимальных фрагментов на заданной фрагментарной структуре. Каждый фрагмент из этого множества определяется как результат работы фрагментарного алгоритма при некоторой заданной перестановке элементарных фрагментов. Таким образом, любому допустимому решению соответствует определенная перестановка чисел 1,2,…, N, где N – количество элементарных фрагментов. Для каждого допустимого решения определено значение целевой функции.

Базовое множество X эволюционной модели – это множество $S_N = \{i_1, i_2, ..., i_N\}$ всех перестановок чисел 1,2,…,N. Оператор построения начальной популяции выделяет подмножество заданной мощности Q из множества X.

Правило вычисления критерия селекции устроено следующим образом: по заданной перестановке фрагментов с помощью фрагментарного алгоритма строится максимальный допустимый фрагмент. Вычисляется значение целевой функции задачи для этого фрагмента.

Опишем теперь оператор кроссовера. Пусть $U = (u_1, u_2, ..., u_N)$ и $V = (v_1, v_2, ..., v_N)$ – две произвольные перестановки. Перестановка-потомок строится следующим образом: последовательности U и V просматриваются с начала. На k-м шаге выбирается наименьший из первых элементов последовательностей и добавляется в новую перестановку-потомок. Затем этот элемент удаляется из двух последовательностей-родителей. Например,

$$K((2,3,4,7,8,1,6,5),(3,4,6,2,1,5,8,7)) = (2,3,4,6,1,5,7,8).$$

Оператор мутации M выполняет случайную транспозицию (замену местами двух элементов) в перестановке.

Оператор селекции выбирает случайным образом набор пар из заданного числа пар во множестве перестановок текущей популяции.

Оператор эволюции элементы промежуточной популяции упорядочивает в последовательность по убыванию значения критерия селекции. В качестве новой текущей популяции выбираются первые Q элементов последовательности.

Обычное правило остановки – количество поколений достигло предельной границы L. Лучшая по значению критерия селекции перестановка из последней построенной популяции определяет приближенное решение задачи.

Предложенный подход является универсальным и позволяет применять один и тот же эволюционный алгоритм к любым оптимизационным задачам на конечных фрагментарных структурах.

Результаты тестирования

Разумным представляется сравнение работы ЭВФ-алгоритма с другими известными алгоритмами. Конечно, хотелось бы получить численные оценки сходимости алгоритма, сравнение по скорости и точности получаемого решения. Однако, как правило, для NP-трудных задач, в которых применяется ЭВФ-алгоритм, это не представляется возможным.

Описание эксперимента.

Входными параметрами при описании серии случайных задач являются: число переменных n; число ограничений m; диапазон изменения коэффициентов $[d_1,d_2]$ в целевой функции и ограничениях задачи; количество задач в серии S.

С помощью генератора случайных чисел генерируется набор коэффициентов c_i, a_{ij},b_j в условии задачи о рюкзаке.

Рассматривались 4 серии задач. Серия А – задачи малой размерности с числом переменных n=10, и с числом ограничений m=6. Серия Б – с числом переменных 20 и количеством ограничений 12. Серия В – задачи с числом переменных 50 и количеством ограничений 30. Серия Г – задачи с числом переменных 50 и количеством ограничений 30. В каждой серии генерировалось 100 задач.

Задачи решались с помощью известного приближенного алгоритма – упорядочения по максимальной удельной ценности, методом случайного поиска и ЭВФ-алгоритмом.

Сравнение алгоритмов осуществлялось по следующим направлениям:

Рекорд – количество задач в серии, где алгоритм оказывался лучшим среди тестируемых. Рейтинг по Борда – сумма числа баллов, набранных на каждой задаче серии. За первое место в сравнении назначалось 5 баллов, за второе 4, за третье 3.

Результаты сравнения алгоритмов на сериях задач приведены в нижеследующих таблицах.

Таблица 1 – Результаты применения различных подходов

Серия	**Число задач**	**Прибл.алгоритм**		**Случ.поиск**		**ЭВФ-алгоритм**	
		Рекорды	**Рейтинг**	**Рекорды**	**Рейтинг**	**Рекорды**	**Рейтинг**
А 10х6	100	44	277	100	400	100	400
Б 20х12	100	10	221	54	348	100	400
В 50х30	100	11	280	0	204	91	391
Г 100х60	100	48	347	0	201	54	354

Выводы

Теоретические результаты и результаты численных экспериментов показывают, что ЭВФ-алгоритм может достаточно эффективно использоваться как эвристический алгоритм в при решении оптимизационных задач линейного булева программирования. ЭВФ-алгоритм является управляемым и качество его работы может возрастать при изменении ряда параметров алгоритма, таких как, величина популяции, число пар для селекции, количество поколений, количество эволюций и т.д.

Литература

1. Козин И.В. Фрагментарные алгоритмы в системах поддержки принятия решений / И.В.Козин // Питання прикладної математики і математичного моделювання, збірник наукових праць. ДНУ : Дніпропетровськ, 2006. – С. 131-137.
2. Козин И.В. Фрагментарный алгоритм для задачи симметричного размещения / И.В.Козин // Радиоэлектроника, информатика, управление.2005, №1. — С. 76-83.
3. Сигал И.Ч. Введение в прикладное дискретное программирование / И.Ч. Сигал, А. П. Иванова. – М. : Физматлит, 2002. – 240 с.
4. Пападимитриу Х. Комбинаторная оптимизация. Алгоритмы и сложность / Х. Пападимитриу, К. Стайглиц. – М. : Мир, 1985. – 512 с.
5. Holland J.H. Adaptation in Natural and Artificial Systems / J.H. Holland. – Boston, MA : MIT Press. – 1992. – 288p.
6. Курейчик В.М. Генетические алгоритмы. Состояние. Проблемы. Перспективы / В.М. Курейчик // Известия РАН. ТиСУ. – 1999. – № 1. – С. 144-160.
7. Скобцов Ю.А. Основы эволюционных вычислений : учеб. пособ./ Ю.А.Скобцов. – Донецк : [ДонНТУ], 2008. – 326 с.

Literatura

1. Kozin I.V. Pitannya prikladnoyj matematiki i matematichnogo modelyuvannya. 2006. № 2. S. 131-137.
2. Kozin I.V. Radioyelektronika, informatika, upravlenie. 2005. № 1. S. 76-83.
3. Sigal I.Ch., Ivanova A.P. Fizmatlit. 2002.
4. Papadimitrou H., Stayglic. Mir. 1985.
5. Holland J. H. Boston, MA : MIT Press. 1992.
6. Kureychik V.M. Izvestiya RAN. TiSU. 1999. №1. S. 144-160.
7. Skobcov Yu.A. Doneck: [DonNTU], 2008.

RESUME

I.V. Kozin

Evolutionary Models Boolean Programming Problem

We study well-known problem of Boolean programming - knapsack problem. This problem has numerous applications in engineering and economics.

To find the approximate solutions of the knapsack problem is proposed to use the evolutionary-fragmentary model.

The paper describes in detail the concept of a fragmented structure. Some properties of fragmentary structures are proved. Fragmentary algorithm, that allows for an order of elementary fragments to construct a maximal allowable fragment of a fragmentary structure, is described.

Shown that the knapsack problem can be seen as a problem with the fragmentary structure. To search for the approximate optimal solution of this problem is proposed evolutionary algorithm, for which the base set is the set of permutations of elementary fragments. The numerical experiment on a large number of test problems showed the effectiveness of the evolutionary model for a search of the optimal solution of the problem of Boolean programming.

Статья поступила в редакцию 02.11.2012.